Diseño de la Base de Datos

Conceptos Básicos

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados para la base de datos, parece lógico emplear el tiempo que sea necesario en aprender los principios de un buen diseño ya que, en ese caso, es mucho más probable que la base de datos termine adaptándose a sus necesidades y pueda modificarse fácilmente.



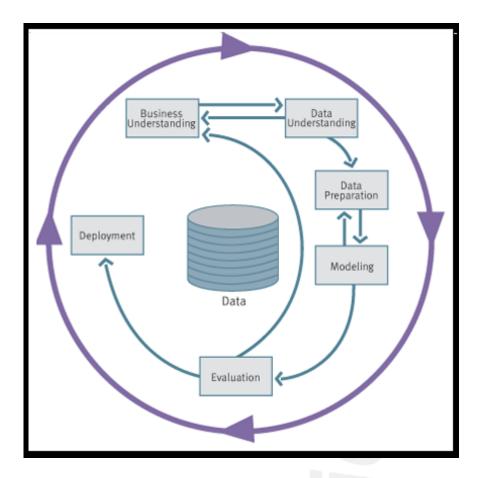
¿Qué es un buen diseño de base de datos?

El proceso de diseño de una base de datos se guía por algunos principios. El primero de ellos es que se debe evitar la información duplicada o, lo que es lo mismo, los datos redundantes, porque malgastan el espacio y aumentan la probabilidad de que se produzcan errores e incoherencias. El segundo principio es que es importante que la información sea correcta y completa. Si la base de datos contiene información incorrecta, los informes que recogen información de la base de datos contendrán también información incorrecta y por tanto, las decisiones que tome a partir de esos informes estarán mal fundamentadas.

Modelo de Datos Relacional

Introducción

Este modelo es utilizado en la confección de las bases de datos y muchas herramientas aplican su teoría. Cuando se encara el proceso de diseño de una base de datos



Recolección de Datos

Lo primero es ocuparse de la recolección de datos y el análisis de requerimientos, etapa durante la cual la persona encargada del diseño visita a los futuros usuarios de la base de datos para así poder conocer cuales son las características de lo que espera el usuario de la base de datos. En esta etapa también deben especificarse los requerimientos funcionales, los cuales consisten en las operaciones definidas por el usuario o transacciones que se aplicarán a la base de datos que incluyen la actualización y la consulta de los datos.

Esquema Conceptual

El siguiente paso es la creación del llamado Esquema Conceptual. Este esquema representa la descripción de los requerimientos del usuario incluyendo características de los datos, las restricciones y sus relaciones, este esquema no tiene características de implementación, generalmente son más fáciles de entender y por este motivo es muy útil para poder comunicarse con usuarios que no son técnicos.



Este esquema conceptual sirve para controlar que todos los requerimientos solicitados por el usuario estén incluidos.

Diseño Lógico de la Base de Datos

El siguiente paso es construir el llamado Diseño Lógico de la Base de Datos el cual consiste en implementar la base

de datos en un SGBD (Sistema de gestión de bases de datos) comercial.

Diseño Físico de la Base de Datos

El paso final es el Diseño Físico de la Base de Datos el cual consiste en determinar las estructuras de almacenamiento internas y la organización de los archivos de la base de datos.





Modelo Entidad Relación

El Modelo Entidad Relación describe los datos mediante entidades, atributos y relaciones.

Entidades y atributos

Una Entidad es algo que representa un objeto y que interesa al sistema.

Un atributo representa lo que describimos de la entidad. Una entidad puede ser básicamente algo tangible como por ejemplo un libro, un teléfono, una mesa, una locomotora, etc., o puede ser algo de existencia conceptual o un rol como ser una empresa, una sucursal, un alumno, un profesor, un cliente, un médico, un contribuyente, etc.

Cada entidad tiene una serie de atributos que la describen, por ejemplo la entidad libro tiene como atributos el ISBN (International Standard Book Number) es el número por el cual se identifica a cualquier libro en el mundo, el nombre del libro, el o los autores, género, cantidad de páginas, editorial, idioma original, edición, etc. La entidad empresa tiene como atributo el nombre, su número de CUIT, su número de ingresos Brutos, dirección, teléfonos, localidad, e-mail, etc. Una entidad en particular tendrá valores para cada uno de sus atributos.

En el siguiente ejemplo observamos dos entidades y posibles valores asociados.

Libros:

ISBN: 968-444-399-4

Nombre: SISTEMAS DE BASES DE DATOS

Autor: Rames A Elmasri, Shamkant B

Navathe

Idioma original: Inglés.

Editorial: Prentice Hall.

Alumnos:

Apellido: De la Sierra

Nombre: Antonio Raúl

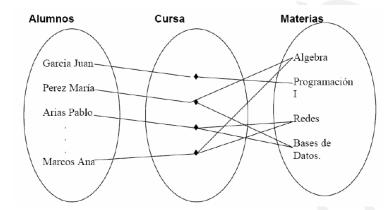
Fecha de Nacimiento: 10/02/1955.

Tipo Documento: DNI

Número Documento: 10.432.987

Relaciones

Otros de los elementos del modelo relacional son las relaciones. Definimos como relación a una abstracción que muestra la asociación entre entidades. Una entidad puede estar conectada a una o más relaciones pero nunca conectada directamente a otra entidad. Veamos un ejemplo: Representamos a los alumnos que cursan materias



Propiedades de una Relación

Las relaciones poseen una serie de propiedades que debemos tener en cuenta.

Ellas son:

- Grado
- Conectividad
- Condicionalidad

Grado de una Relación

Definimos como grado de una relación a la cantidad de entidades que forman parte de la relación, así encontramos a relaciones de diferentes grados:

Unario en donde una sola entidad forma parte de la relación.

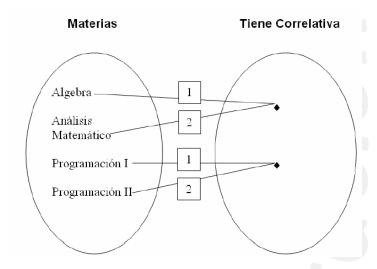
Binario dos entidades forman parte de la relación.

Ternario tres entidades forman parte de la relación.

Estudiemos cada uno de los casos: los dos ejemplos anteriores tienen una relación de grado binario pues en el primero intervienen dos entidades (Alumnos y Materias) y en el segundo (Empleados y Sectores de una Empresa), de este tipo de grado son la gran mayoría de las relaciones de un modelo relacional.

Como ya dijimos cuando el grado de una relación es unario, sólo una entidad forma parte de la relación, estos casos no son muy comunes.

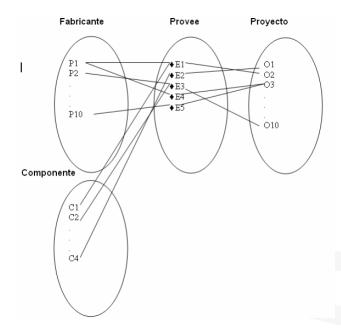
Un ejemplo típico es la relación de una materia con ella misma con respecto a la correlatividad, por ejemplo la materia Programación II tiene como correlativa a Programación I, es decir que una materia está relacionada con otra materia de la misma entidad.



En el ejemplo anterior tenemos Álgebra como materia correlativa de Análisis Matemático, y a Programación I como correlativa de Programación II.

Otro ejemplo es el de las estaciones de subte en donde una estación es precedida por otra estación, en este caso también una estación de la entidad estaciones está relacionada con otra estación de la misma entidad.

En las relaciones de grado ternario son tres las entidades que forman parte de la relación, este tipo de relación así como las unarias no son muy comunes, en donde cada una de las tres entidades está relacionada con las otras dos. Un ejemplo puede ser la relación que existe entre un fabricante un componente y un proyecto, veamos este ejemplo en forma abstracta.

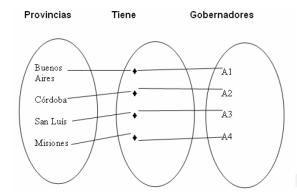


Conectividad de una Relación

Por conectividad entendemos a la cantidad de instancias de un objeto que participan en la relación con otro objeto

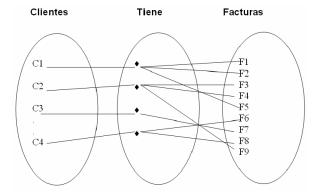
Relación 1:1

Veamos un ejemplo de una relación de 1:1, con la relación que existe entre una provincia y un gobernador, en este caso una Provincia tiene un solo gobernador y un gobernador gobierna una sola Provincia.



Relación 1:M

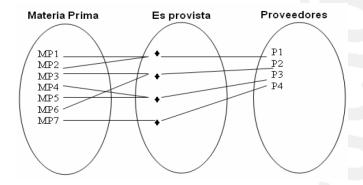
En este caso una instancia de una entidad se relaciona con una o muchas instancias de la otra entidad, veamos un ejemplo:



Veamos que en el ejemplo anterior el cliente C3 está relacionado sólo con una factura, es decir que cuando hablamos de muchos es el máximo de instancias con el cual una instancia se relaciona con otra u otras instancias de la otra entidad.

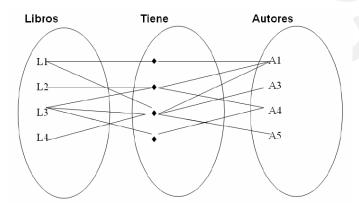
Relación M:1

Una relación de tipo muchos a uno es similar a la del ejemplo anterior todo depende del lugar en el cual se encuentra la entidad, veamos un ejemplo en donde supuestamente un proveedor provee una o más materias primas



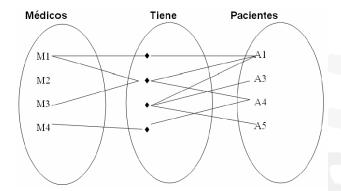
Relación M:N

En este tipo de relación cada una de las instancias de la primera entidad se relaciona con una o muchas instancias de la otra entidad y viceversa es decir cada una de las instancias de la segunda entidad se relaciona con una o muchas instancias de la primera entidad, en el siguiente ejemplo tenemos las entidades libros y autores en la cual un libro es escrito por uno o muchos autores y por el otro lado un autor lo es de uno o muchos libros.



Condicionalidad de una relación

Con la propiedad condicionalidad estamos indicando si la relación entre las entidades es obligatoria u opcional, lo que significa determinar si existen o no instancias de ambas entidades que forman parte de la relación. Supongamos el siguiente ejemplo: las entidades son profesor y materia la relación es directa, en la cual una materia puede no tener asignado ningún profesor, y un profesor puede dictar varias materias. Otro ejemplo puede ser en la relación pacientes y médicos en una clínica un médico puede no atender a ningún paciente ya que puede ser el director de la clínica.



Del ejemplo anterior observamos que si bien la relación es de cardinalidad M a N, el médico M2 no atiende a ningún paciente por lo tanto esta relación es condicional.

Diseño Físico de la Base de Datos

Definición

El diseño físico es el proceso de traducción del modelo lógico abstracto a un diseño técnico específico para el nuevo sistema.

Entonces el Diseño Físico de la Base de Datos: Es el proceso de producción de una descripción, de una implementación, de un almacenamiento secundario de la Base de Datos, describe el almacenamiento de estructuras y métodos de acceso usados para conseguir el acceso eficiente a los datos.

Requisitos

El Diseño Físico de la Base de Datos es la última etapa del proceso de diseño, en el cual, teniendo presentes los requisitos de los procesos, características del SGBD, del SO y el hardware, se pretenden los siguientes objetivos:

Disminuir los tiempos de respuesta.

Minimizar espacio de almacenamiento.

Evitar las reorganizaciones.

Proporcionar la máxima seguridad.

Optimizar el consumo de recursos.

Diagrama Entidad Relación (DER)

Introducción

El Modelo Relacional cuenta básicamente con dos herramientas: el Diagrama Entidad Relación, conocido con el nombre de DER y el Diccionario de Datos del Modelo Relacional. El Diagrama Entidad Relación es una herramienta gráfica que nos permite representar el Modelo Relacional con todas sus entidades y sus relaciones. Con el Diagrama Entidad Relación se observan los datos almacenados en el sistema y es representado como una red de entidades conectadas con relaciones.

Elementos de un Diagrama Entidad Relación

Entidades

Las entidades son uno de los elementos que forman parte de un DER, una entidad es representada con un rectángulo donde en su interior se escribe el nombre de la entidad, según el autor que se consulte, en el DER pueden o no estar incluidos los atributos de la entidad

Ejemplos:



Clientes = Numero de cliente, Apellido, Nombre, Domicilio, Código Postal, Localidad, CUIT, Número de Ingresos Brutos, Telefono, dirección e-mail.

Relaciones

Las relaciones son otro de los elementos que forman parte de un DER, y son representados mediante un rombo, donde en su interior se escribe el nombre de la relación, para este nombre se utiliza o un verbo que representa la asociación entre las dos entidades que relaciona o el nombre de las dos entidades separadas por un guión. Veamos un ejemplo. Para una relación que conecta Médicos con Pacientes. Las dos opciones son las siguientes:

1)_ Utilizando el verbo Tiene como nombre de la relación.



2) Médicos - Pacientes.

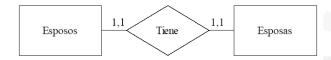


Estudiemos detenidamente la relación Pacientes Médicos. De acuerdo con lo visto más arriba esta es una relación de grado binario pues la relación está compuesta por dos entidades y con respecto a su cardinalidad es de tipo M:N (o de Muchos a Muchos). Lo representamos de la siguiente manera:



Se lee de la siguiente manera: un médico tiene uno o muchos pacientes (N) y un paciente tiene uno o muchos médicos (M).

Para una relación de 1:1



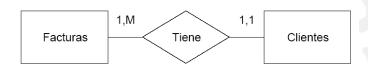
En este caso decimos que un esposo está relacionado como mínimo con una esposa y como máximo con una esposa (por lo menos en nuestra cultura occidental).

Para una relación de 1:M



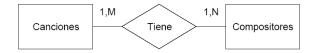
Significa que un empleado trabaja en un proyecto como mínimo y un proyecto como máximo y en un proyecto trabajan uno o muchos empleados.

Para una relación de M:1



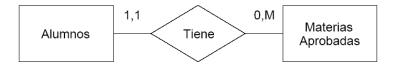
Una factura le pertenece a un cliente como mínimo y a un cliente como máximo, y un cliente tiene una factura como mínimo y muchas facturas como máximo.

Para una relación M:N.



Veamos ahora ejemplos con relaciones condicionales.

Para el caso 1,M condicional



En donde un alumno no tiene aprobada ninguna o muchas materias, ninguna puede ser para el caso de un alumno que recién se inscribe en primer año y todavía no aprobó ningún final.

Para el caso 1,1 condicional.



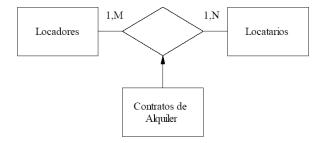
Donde un empleado puede o no tener una Tarjeta de Crédito Corporativa y una Tarjeta de Crédito Corporativa le pertenece a un solo empleado de la empresa.

Tipos de Objeto Asociativo

Otro de los elementos que forman parte de un Diagrama Entidad Relación son los llamados Tipos de Objeto Asociativo o TOA, estos tipos de elementos aparecen cuando en una relación además de conectar las entidades posee atributos propios, es decir que contiene información que existe siempre y cuando exista la relación y en el caso que la relación no exista, tampoco existirá el TOA.

Un Tipo de Objeto Asociativo es representado en un Diagrama Entidad relación como un rectángulo que está conectado a través de una flecha a la relación.

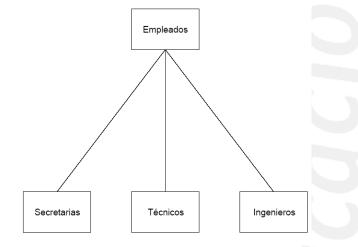
En el ejemplo siguiente tenemos las entidades locador y locatario que está conectada por una relación, de esa misma relación surge el contrato de alquiler que representa el Tipo de Objeto Asociativo.



Supertipo, Subtipo y Especialización

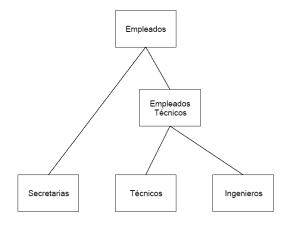
En algunos casos una entidad tiene varias subagrupaciones adicionales que deben representarse explícitamente por su importancia en la base de datos.

El conjunto de instancias de cada una de estas agrupaciones es un subconjunto de la entidad. Por ejemplo dada la entidad empleado existen diferentes subgrupos: secretarias, técnicos, administrativos, etc., lo que significa que toda entidad que sea miembro de una de estas subagrupaciones también será un empleado.



Podemos decir que un subtipo hereda todos los atributos del supertipo.

Especialización. La especialización es el hecho de agrupar un conjunto de subtipos de un supertipo. Este agrupamiento se produce teniendo en cuenta alguna característica distintiva del subtipo.



En este caso Empleados técnicos es una especialización de los subtipos técnicos e ingenieros.

Técnica para construir un DER

Para construir un DER puede utilizarse como regla de práctica básica el tomar los sustantivos importantes como entidades y los verbos importantes como relaciones:

- 1) Identificar las entidades.
- 2) Individualizar identificadores únicos de las entidades, si la entidad posee una sola instancia, este elemento no es una entidad.
- 3) Identificar relaciones entre objetos.
- 4) Clasificar relaciones según conceptos de grado, conectividad, condicionalidad.
- 5) Identificar tipos de objetos asociativos.
- 6) Identificar supertipos y subtipos, agrupando objetos que posean atributos comunes y alguna condición de diferenciación.
- 7) Dibujar el DER según la notación de la herramienta.
- 8) Eliminar elementos redundantes o fuera del alcance del sistema.

Reglas de construcción de un DER

- 1) Una entidad puede estar conectada a una o más relaciones.
- 2) Una relación debe conectarse a una o más entidades.
- 3) Una entidad no puede estar conectada directamente a otra entidad.
- 4) Una relación no puede estar conectada directamente a otra relación.
- 5) No pueden existir distintos elementos con el mismo nombre.
- 6) No incluir relaciones irrelevantes para el sistema.
- 7) Eliminar relaciones que no puedan existir en el mundo real.
- 8) Eliminar relaciones redundantes.
- 9) Si una entidad sólo tiene su identificador como atributo, eliminarla e incluir la información en otra entidad.

Veamos ahora dados algunos enunciados ejemplos de Diagramas Entidad Relación.

Ejemplo Integrador

Empresa Constructora de Edificios

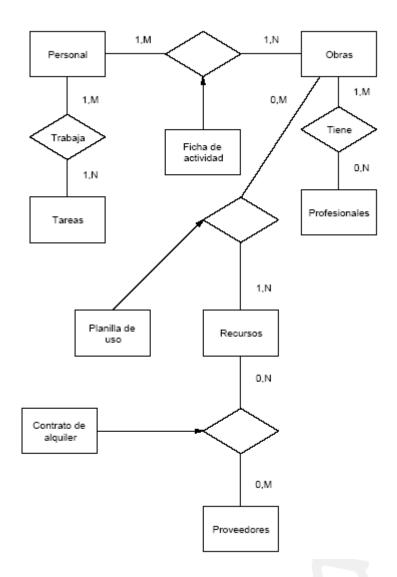
Esta empresa cuenta con una nómina de personal disponible para poder ser asignado a las diferentes obras. Una persona puede estar asignada a una o más obras, y en una obra pueden trabajar una o más personas, de esta asignación debe aparecer una ficha de actividad de las personas en las obras, en donde debe figurar la fecha, la hora y la actividad realizada.

Toda persona está especializada en una o más tareas, y una tarea puede ser realizada por una o más personas de la empresa (ejemplos frentista, colocador de azulejos, electricistas, yeseros, etc.).

Toda obra además del personal propio puede tener asignado profesionales para realizar aquellas tareas con la que no existe personal especializado (como es el caso de algún ingeniero que estudie el suelo para saber si este puede resistir los pisos que se desean construir, o el impacto ambiental de la obra, etc.), por este motivo un profesional puede estar asignado a una o más obras y una obra puede contar con la colaboración de ninguno o muchos profesionales.

Toda obra necesita contar con una serie de recursos como ser mezcladoras de cemento, martillos neumáticos, inyectores de hormigón, etc. y un recurso puede ser usado en ninguna o muchas obras. Por cada recurso usado en cada obra debe figurar una planilla de uso del recurso, en donde se debe almacenar las fechas en el que fue utilizado y si sufrió roturas.

Los recursos pueden pertenecer o no a la empresa constructora, en el caso que no lo sean, estos se pueden alquilar a un proveedor, por lo cual se deberá confeccionar un contrato de alquiler.



DICCIONARIO DE DATOS

Vamos a utilizar el diccionario de datos para describir los elementos que forman parte de un DER.

Agregamos una @ a los atributos que sirvan como identificación única.

Objetos = @identificador + atributos.

Tipo de objeto asociativo = {@identificador1} + {@identificador2} + atributos.

Relación

```
@identificador1 + @identificador2 (caso 1:1).
{@identificador1} + @identificador2 (caso N:1).
{@identificador1} + {@identificador2} (caso m:n).
Super tipo = {atributos comunes} + {subtipos}.
Sub tipo = idem objetos.
```

Entidades

PERSONAL = @Nro_Legajo + Apellido + Nombre + Tipo_Documento + Nro_Documento + Sexo + Domicilio + Localidad + Codigo_Postal + Fecha_Nacimiento + Telefono.

OBRAS = @Nro_Obra + Denominación + Direccion + Localidad + Nro_Permiso.

PROFESIONALES = @Nro_CUIT + Apellido + Nombre + Domicilio + Localidad + Localidad + Codigo_Postal + Telefono + Tipo_Documento + Nro_Documento + Fecha_Nacimiento

TAREAS = @Codigo_Tarea + Denominación + Caracteristicas + Observaciones.

RECURSOS = @Codigo_Recurso + Denominación + Caracteristicas.

PROVEEDORES = @Nro_CUIT + Razon_Social + Telefono + Direccion + Localidad + Codigo_Postal + Email.

Tipos de Objeto Asociativo

FICHA_DE_ACTIVIDAD = @Nro_Legajo + @Nro_Obra + Fecha + Hora + Actividad_Realizada.

PLANILLA_DE_USO = @Nro_Obra + @Codigo_Recurso + Fecha + Roturas

CONTRATO_DE_ALQUILER = @Codigo_Recurso + @Nro_CUIT + Fecha_Inicio_Contrato + Fecha_Fin_Contrato + Monto_Alquiler

Relaciones

TRABAJA = {@Nro Legajo} + {@Codigo Tarea}

 $TIENE = \{ @Nro_Obra \} + \{ @Nro_CUIT \}$

Normalización

Definición

Una vez realizada la implementación del Diseño Físico de la Base de Datos viene el proceso de Normalización de la Base de Datos.

La Normalización es un conjunto de reglas y procedimientos que permiten definir una estructura de datos simple y estable frente a los cambios.

Trabajamos con la siguiente estructura:

Factura = @#factura + #cliente + nombre cliente + fecha + {#artículo + nombre artículo + cantidad artículo + precio unitario + precio por cantidad} + precio total.

Analizándola observamos que por cada #factura aparecen uno o muchos artículos, es decir como pasa realmente, cuando realizamos una compra, por cada factura figuran uno o varios artículos que compramos junto con la cantidad de artículos, el precio unitario y el precio por la cantidad.

Decimos entonces que @articulo, nombre articulo, cantidad artículo, precio unitario y precio por cantidad forman un grupo repetitivo.

Veamos las formas normales

PRIMERA FORMA NORMAL

Una estructura de datos sin grupos repetitivos está automáticamente en Primera Forma Normal.

Técnica de Construcción.

Veamos cómo transformamos una estructura para que quede en primera forma normal

- 1) Identificar claves únicas.
- 2) De las claves únicas, seleccionar la clave primaria.
- 3) Dividir el objeto en tantos objetos como sean necesarios para separar los grupos repetitivos.

Volvamos al ejemplo, de la estructura original FACTURA, y la transformamos en este caso en dos estructuras:

Factura_Cabecera = @#factura + #cliente + nombre cliente + fecha + precio total.

Factura_Detalle = @#factura + @#artículo + nombre artículo + cantidad artículo + precio unitario + precio por cantidad.

En donde observamos que ya no existen grupos repetitivos, continuamos trabajando pero ahora con FACTURA CABECERA y FACTURA DETALLE.

SEGUNDA FORMA NORMAL

Un objeto está en Segunda Forma Normal si está en Primera Forma Normal y además todos los atributos no clave tienen dependencia funcional completa con la clave primaria.

Técnica de Construcción

Si la clave primaria está compuesta por más de un atributo, verificar que cada atributo que no pertenece a la clave primaria depende de toda la clave y no solamente de una parte. Dividir el objeto si es necesario para lograr este objetivo. Si la clave primaria está compuesta de un solo atributo la estructura ya se encuentra en segunda forma normal.

Factura_Cabecera = @#factura + #cliente + nombre cliente + fecha + precio total.

Factura_Detalle = @#factura + @#artículo + cantidad artículo + precio por cantidad.

Articulo = @#artículo + nombre artículo + precio unitario.

Observamos que en factura detalle no es suficiente el #factura para identificar la clave primaria, por eso ahora la clave primaria está compuesta por #factura y #artículo.

TERCERA FORMA NORMAL

Un objeto está en Tercera Forma Normal si está en Segunda Forma Normal y además todos sus atributos que no son clave primaria sean independientes entre sí.

Técnica de Construcción

Verificar que todos los atributos no clave sean mutuamente independientes entre sí. Suprimir atributos redundantes o dividir los objetos.

Continuamos trabajando con Factura Cabecera, Factura Detalle y Artículos.

Ejemplo:

Factura_Cabecera = @#factura + #cliente + fecha.

Cliente = @#cliente + nombre cliente.

Factura Detalle = @#factura + @#artículo + cantidad artículo.

Articulo = @#artículo + nombre artículo + precio unitario.

Nota: precio total y precio por cantidad se deben eliminar ya que se pueden calcular con otros atributos. Precio por cantidad multiplicando la cantidad de artículos por el precio unitario, y el precio total con la suma de todos los precios por cantidad.

DIAGRAMA DE TABLAS

Cuando estudiamos el Diagrama Entidad Relación, dijimos que representaba la relación entre diferentes objetos como son las entidades, tipos de objeto asociativo, supertipos y subtipos, ya conocimos que significa la normalización y cómo las relaciones del Diagrama Entidad Relación son representadas físicamente en tablas, emplearemos todos estos conocimientos para construir un Diagrama de Tablas. El Diagrama de Tablas también conocido como Diagrama de Datos en inglés Data Diagrammer, representa a las tablas relacionadas. Veamos los pasos a seguir:

- 1) Construir el Diagrama Entidad Relación junto con su Diccionario de Datos asociado.
- 2) Aplicar la técnica de normalización a todos los elementos del Diagrama Entidad Relación.
- 3) Transformamos estos nuevos objetos considerando su representación física.
- 4) Construimos el Diagrama de Tablas.
- 5) Definimos los tipos de datos y su longitud para cada unos de los campos que forman parte de las tablas.

Veamos un ejemplo:

Continuamos con el ejemplo visto de Armado de Televisores.

Ahora normalizamos

Operarios se encuentra en primera, segunda y tercera forma normal.

Televisores se encuentra en primera, segunda y tercera forma normal.

Importador se encuentra en primera, segunda y tercera forma normal.

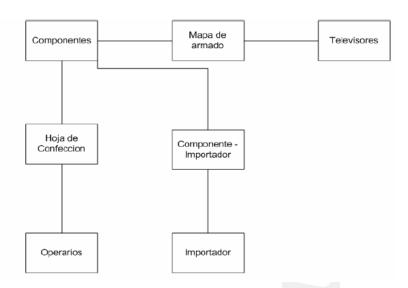
Componentes se encuentra en primera, segunda y tercera forma normal.

Mapa_de_Armado se encuentra en primera, segunda y tercera forma normal.

Hoja_de_Confección se encuentra en primera, segunda y tercera forma normal.

En este caso no debemos normalizar ninguna estructura, es común que en la realidad suceda esto ya que al conocer normalización se tiende a diseñar las entidades ya normalizadas.

Diagrama de Tablas



Ahora vemos la composición de las tablas.

Operarios

Nro_legajo int not null

Apellido char(30) not null

Nombre char(30) not null

Domicilio char(25)

Telefono char(15)

Tipo_doc char(3) not null

Numero_doc int not null Nro_legajo clave primaria Televisores Cod_Televisor int not null Modelo char(25) not null Caracteristicas char(50) Cod_Televisor clave primaria Importador Nro_importador int not null Razon_social char(30) not null CUIT number (20) not null Nro_importador clave primaria Componentes Cod_componente int not null Descripción char(50) Cod_componente clave primaria Mapa_de_Armado Cod_componente int not null Cod_televisor int not null Cantidad int Ubicación char(50) Cod_componente y Cod_televisor son clave primaria. Hoja_de_Confeccion

Cod_Confeccion int not null Cod_componente int not null Nro_Legajo int not null

Cod_Confeccion es clave primaria.

Componente_Importador

Cod_Relacion int identity (1,1) not null

Cod_componente int not null

Nro_importador int not null

Cod_Componente y Nro_Importador son clave primaria

